What is claimed is:

1. A method of assembling a source code module to form an object code module; said source code module including one or more assembler directives, wherein the assembler directives are used to generate relocation instructions in the object code module, the method comprising:

reading a plurality of compound relocation sequence definitions stored in a memory, each compound relocation definition sequence comprising a compound relocation indicator and a first sequence of relocation instructions;

reading assembler source code from said source code module, said source code generating an associated sequence of relocation instructions for executing the directive;

determining if said associated sequence of relocations matches one of said stored sequence of relocation instructions; and

if a match of relocation sequences is determined, inserting into said object code module a compound relocation including the compound relocation indicator of said matched compound relocation sequence definition instruction and said matched sequence.

2. A method according to claim 1, wherein the compound relocation instruction is inserted into at least one relocation section of the object code module associated with a set of section data, said compound relocation instruction defining an offset indicating where in the section data the compound relocation instruction is to be effected at link time.

3. A method according to claim 1, wherein said compound relocation instructions are inserted into relocation sections associated respectively with sets of section data, each compound relocation instruction defining an offset indicating where in the section data the compound relocation instruction is to be effected at link time.

4. A method according to claim 1, wherein the compound relocation definition instruction is inserted into the object code module in a relocation definition section.

5. A method according to claim 1, wherein at least one of the relocation instructions of said sequence of relocations associated with the assembler directive has at least one argument field for holding a parameter and said assembler directive has at least one argument field for holding a parameter, said method further comprising:

copying said parameter held in said argument field of said assembler directive into said argument field of said compound relocation instructions.

6. A method according to claim 1, wherein said compound relocation sequence definition instructions are read from an include source code file in response to a further assembler directive being read by said assembler.

7. A method according to claim 6 further comprising reading the status of an interoperability flag and selectively carrying out the method of any preceding claim or inserting into the object code module said sequence of relocation instructions associated with said assembler directive.

8. A method according to claim 7, wherein said interoperability flag is set by said assembler in response to receiving said further assembler directive.

9. A method of generating an include source code file for use with the method of claim 6 comprising:

inputting one or more special assembler directives to an assembler, each special directive identifying an assembler source code having an associated sequence of relocation instructions for which it is desired to generate a compound relocation sequence definition;

in response to one of said special directives, generating a compound relocation sequence definition comprising a compound relocation indicator and said associated sequence of relocation instructions;

outputting said generated compound relocation sequence definition from said assembler to a special object code module;

inputting said special object code module to a lister, said lister converting said compound relocation sequence definitions to linker insert instructions and outputting said linker insert instructions to generate said include source code module.

10. A method according to claim 9, wherein the step of generating the compound relocation definition instruction comprises:

generating said compound relocation indicator having a field for holding a value denoting the number of relocation instructions in said sequence and setting said field value at zero;

counting the number of relocation instructions in said associated sequence; and

inserting said counted number into said field of the compound relocation indicator.

11. An assembler for assembling an object code module from a source code module comprising program instructions, data and assembler directives, that are executable to generate relocation instructions in the object code module, the assembler comprising:

a source code reader for reading said program instructions and directives from the source code module; and

a directive translator for generating a compound relocation definition instruction in response to said read assembler directive and an associated sequence of relocation instructions said compound relocation definition instruction defining a compound relocation indicator and the number of instructions in said associated sequence of relocation instructions.

12. An assembler according to claim 11 further comprising an interoperability flag and selecting means, said selecting means being responsive to the status of said interoperability flag to select a mode of operation of said assembler wherein in a first mode of operation said directive translator generates said compound relocation definition instruction and in a second mode of operation said directive translator generates only said associated sequence of relocation instructions.

13. An assembler according to claim 11, wherein said directive translator comprises:

a relocation generator for generating said compound relocation definition instruction and said associated sequence of relocation instructions; and

a counter for counting the number of relocation instructions in said associated sequence.

14. An assembler according to any one of claims 11, wherein said source code translator further comprises:

a store for storing at least one parameter held in a respective argument field in said assembler source code; and

a relocation modifier arranged to detect at least one of a symbol or value in at least one of said relocations instructions in said associated sequence of relocation instructions that matches said parameter and to replace said matched symbol or value in said relocation with a reference to said parameter.

15. A toolchain for forming a target executable program from a plurality of source code modules, each source code module comprising program instructions or data and assembler directives, the toolchain comprising:

an include source code module comprising a plurality of compound relocation definition instructions, each compound relocation definition sequence comprising a compound relocation indicator and a first sequence of relocation instructions;

an assembler for assembling said plurality of source code modules and said include source code module to form a plurality of object code modules; and

a linker for linking said object code modules to form the target executable program.